**PL/SQL:**

**EXERCISE:1**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Age NUMBER,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER REFERENCES Customers(CustomerID),

InterestRate NUMBER,

DueDate DATE

);

INSERT INTO Customers VALUES (10, 'Ravi Kumar', 63, 11000, 'FALSE');

INSERT INTO Customers VALUES (11, 'Sneha Roy', 59, 4000, 'FALSE');

INSERT INTO Customers VALUES (12, 'Arjun Mehta', 72, 9500, 'FALSE');

INSERT INTO Customers VALUES (13, 'Priya Singh', 35, 15000, 'FALSE');

INSERT INTO Loans VALUES (201, 10, 6.7, SYSDATE + 20);

INSERT INTO Loans VALUES (202, 11, 7.1, SYSDATE + 10);

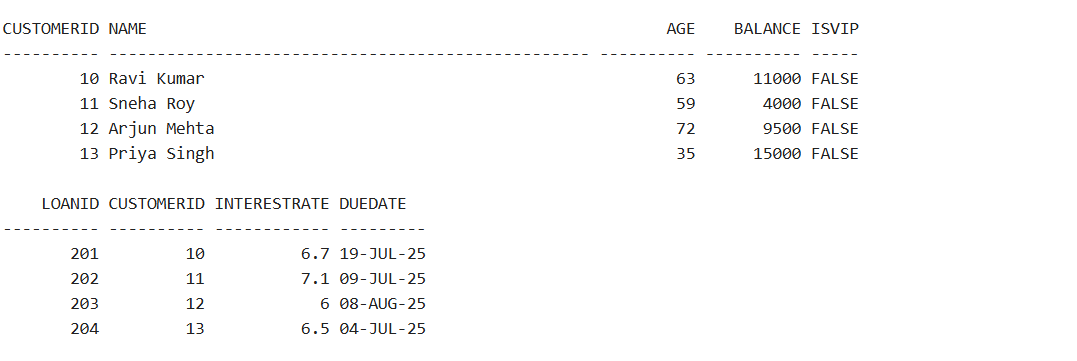
INSERT INTO Loans VALUES (203, 12, 6.0, SYSDATE + 40);

INSERT INTO Loans VALUES (204, 13, 6.5, SYSDATE + 5);

COMMIT;

SELECT \* FROM Customers;

SELECT \* FROM Loans;



**Scenario 1: Interest discount for age > 60**

BEGIN

FOR cust IN (SELECT CustomerID FROM Customers WHERE Age > 60) LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = cust.CustomerID;

END LOOP;

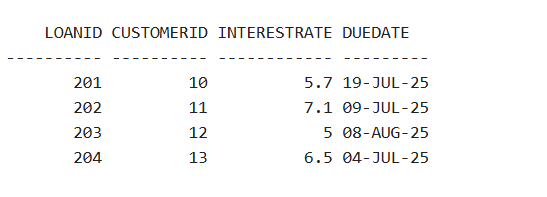
COMMIT;

END;

/

SELECT \* FROM Loans;

**OUTPUT:**



**Scenario 2: Mark VIPs with balance > 10000**

BEGIN

FOR cust IN (SELECT CustomerID FROM Customers WHERE Balance > 10000) LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust.CustomerID;

END LOOP;

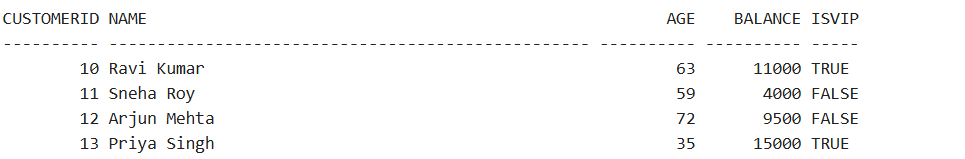
COMMIT;

END;

/

SELECT \* FROM Customers;

**OUTPUT:**



**Scenario 3: Reminders for loans due in next 30 days**

BEGIN

FOR loan\_rec IN (

SELECT c.Name, l.DueDate

FROM Loans l

JOIN Customers c ON c.CustomerID = l.CustomerID

WHERE l.DueDate <= SYSDATE + 30

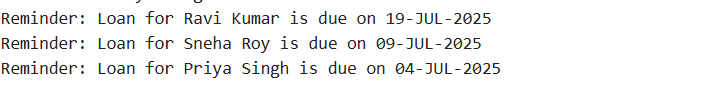
) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan for ' || loan\_rec.Name ||

' is due on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-MON-YYYY'));

END LOOP;

END;

/

**EXERCISE:3**

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerName VARCHAR2(50),

Balance NUMBER,

AccountType VARCHAR2(20)

);

CREATE TABLE Employees (

EmpID NUMBER PRIMARY KEY,

EmpName VARCHAR2(50),

Department VARCHAR2(50),

Salary NUMBER

);

INSERT INTO Accounts VALUES (201, 'Karan Patel', 5000, 'Savings');

INSERT INTO Accounts VALUES (202, 'Divya Sharma', 2500, 'Current');

INSERT INTO Accounts VALUES (203, 'Manoj Reddy', 8000, 'Savings');

INSERT INTO Accounts VALUES (204, 'Neha Joshi', 7000, 'Savings');

INSERT INTO Employees VALUES (11, 'Kavita Nair', 'Finance', 60000);

INSERT INTO Employees VALUES (12, 'Rohan Jain', 'Marketing', 48000);

INSERT INTO Employees VALUES (13, 'Swati Bhat', 'Finance', 62000);

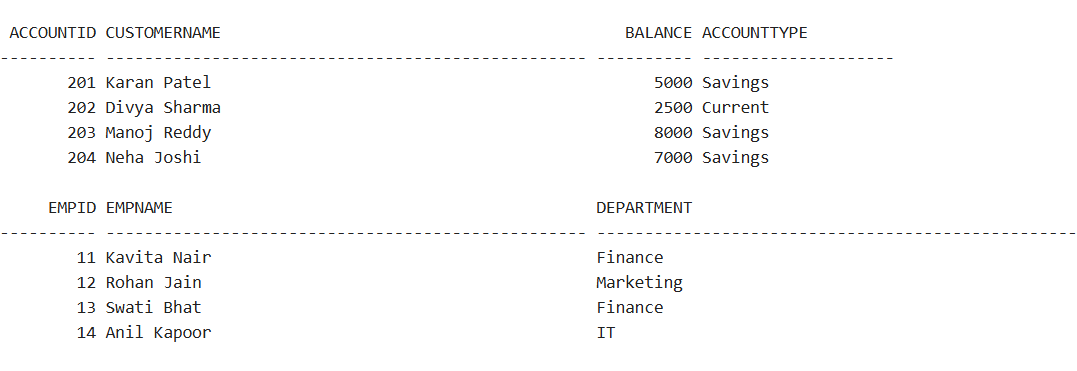
INSERT INTO Employees VALUES (14, 'Anil Kapoor', 'IT', 52000);

COMMIT;

SELECT \* FROM Accounts;

SELECT \* FROM Employees;

**OUTPUT:**



**Scenario 1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance \* 1.01

WHERE AccountType = 'Savings';

COMMIT;

END;

/

BEGIN

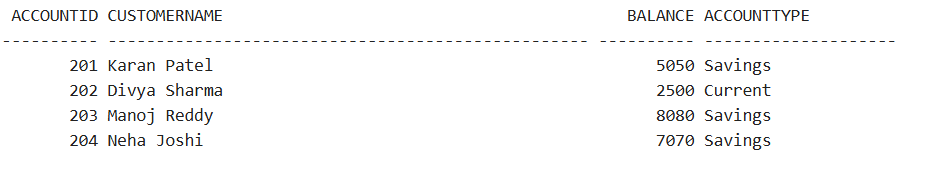
ProcessMonthlyInterest;

END;

/

SELECT \* FROM Accounts;

**OUTPUT:**



**Scenario 2: UpdateEmployeeBonus**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* p\_bonus\_percent / 100)

WHERE Department = p\_dept;

COMMIT;

END;

/

BEGIN

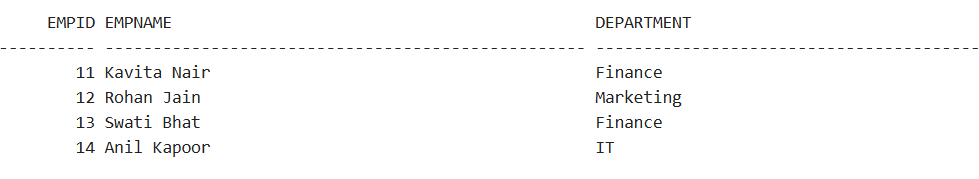
UpdateEmployeeBonus('Finance', 15);

END;

/

SELECT \* FROM Employees;

**OUTPUT:**



**Scenario 3: TransferFunds**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_from\_account;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance for transfer.');

END IF;

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account;

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account;

COMMIT;

END;

/

BEGIN

TransferFunds(204, 202, 1000);

END;

/

SELECT \* FROM Accounts;

**OUTPUT:**

